LAB SESSION 2

SQL DISTINCT

The **SQL DISTINCT** clause is used together with the SQL SELECT keyword, to return a dataset with unique entries for certain database table column.

We will use our Custom database table to illustrate the usage of **SQL DISTINCT**.

FirstName	LastName	Email	DOB	Phone
John	Smith	John.Smith@yahoo.com	2/4/1968	626 222-2222
Steven	Goldfish	goldfish@fishhere.net	4/4/1974	323 455-4545
Paula	Brown	pb@herowndomain.org	5/4/1978	416 323-3232
James	Smith	jim@supergig.co.uk	20/10/1980	416 323-8888

For example if we want to select all distinct surnames from our Custom table, we will use the following **SQL DISTINCT** statement:

```
SELECT DISTINCT LastName
FROM Custom
```

The result of the **SQL DISTINCT** expression above will look like this:

LastName
Smith
Goldfish
Brown

EXERCISE:

1. List down the towns from the customers table without duplication.

   SELECT DISTINCT TOWN FROM CUSTOMERS;

TOWN
Mill Park
Preston
Whittlesea
bundoora

2. When customers made order for the product purchase.

   SELECT DISTINCT CUST_NAME
   FROM CUSTOMERS;

CUST_NAME
Di Hunter
Glads Gladdies
Mill Park
Nevs Nursery
Preston City

3. List down the customers who have a purchase

   SELECT DISTINCT CUST_NO
   FROM ORDERS;

CUST_NO
1066
13144

4. List down the product groups from the product table using distinct.

   SELECT DISTINCT PROD_GROUP FROM PRODUCTS;

P
A
C
D

Using the Where Clause:

EXAMPLES:

The **SQL WHERE** clause is used to select data conditionally, by adding it to already existing SQL SELECT query. We are going to use the **Customers** table from the previous chapter, to illustrate the use of the **SQL WHERE** command.

Table: **Custom**

FirstName	LastName	Email	DOB	Phone
John	Smith	John.Smith@yahoo.com	2/4/1968	626 222-2222
Steven	Goldfish	goldfish@fishhere.net	4/4/1974	323 455-4545
Paula	Brown	pb@herowndomain.org	5/24/1978	416 323-3232
James	Smith	jim@supergig.co.uk	20/10/1980	416 323-8888

If we want to select all customers from our database table, having last name 'Smith' we need to use the following SQL syntax:

```
SELECT *
FROM Custom
WHERE LastName = 'Smith'
```

The result of the SQL expression above will be the following:

FirstName	LastName	Email	DOB	Phone
John	Smith	John.Smith@yahoo.com	2/4/1968	626 222-2222
James	Smith	jim@supergig.co.uk	20/10/1980	416 323-8888

In this simple SQL query we used the "=" (Equal) operator in our WHERE criteria:

LastName = 'Smith'

But we can use any of the following comparison operators in conjunction with the **SQL WHERE** clause:

<> (Not Equal)

```
SELECT *
FROM Custom
WHERE LastName <> 'Smith'
```

> (Greater than)

```
SELECT *
FROM Custom
WHERE DOB > '1/1/1970'
```

>= (Greater or Equal)

```
SELECT *
FROM Custom
WHERE DOB >= '1/1/1970'
```

< (Less than)

```
SELECT *
FROM Custom
```

```
WHERE DOB < '1/1/1970'
```

<= (Less or Equal)

```
SELECT *
FROM Custom
WHERE DOB =< '1/1/1970'
```

LIKE (similar to)

```
SELECT *
FROM Custom
WHERE Phone LIKE '626%'
```

Note the LIKE syntax is different with the different RDBMS (SQL Server syntax used above).

Between (Defines a range)

```
SELECT *
FROM Custom
WHERE DOB BETWEEN '1/1/1970' AND '1/1/1975'
```

Where clause are used to limit the rows selected from a query:

SELECT [column name] FROM [table name] WHERE [conditions]

EXERCISE:

1. View the customer name where the customer number is 1066?
   SELECT CUST_NAME FROM CUSTOMERS
   WHERE CUST_NO = 1066;

CUST_NO	CUST_NAME
1066	Nevs Nursery

2. View the customers staying in town : Bundoora from table customers?

   Select cust_name, town from Customers
   where town = 'bundoora';

CUST_NAME	TOWN
Nevs Nursery	bundoora

3. Select the customers who are staying the town called Mill Park? Calculate the number of customers staying in Mill Park?
   SELECT CUST_NAME, TOWN
   FROM CUSTOMERS
   WHERE TOWN ='MILL PARK';

CUST_NAME	TOWN
Glads Gladdies	Mill Park
Mill Park	Mill Park

4. From table Products, view the product code which are categorized in group A?
   SELECT PROD_COD,PROD_GROUP FROM PRODUCTS
   WHERE PROD_GROUP ='A';

PROD_COD	P
MOO	A
LOO	A

Using Comparison Operators with the Where clause:

1. Identify products which cost $100.00 from table Products?
   SELECT PROD_COD, LIST_PRICE
   FROM PRODUCTS
   WHERE LIST_PRICE = 100;

PROD_COD	LIST_PRICE
LTUB	100

2. What are the products which cost more than $300.00 from table Products?

SELECT PROD_COD, LIST_PRICE
FROM PRODUCTS

WHERE LIST_PRICE > 300;

PROD_COD	LIST_PRICE
LTANK	450
GABBY	500

3. Identify the products which cost more than or equal to $100.00?
   SELECT PROD_COD, LIST_PRICE
   FROM PRODUCTS
   WHERE LIST_PRICE >= 100;

PROD_COD	LIST_PRICE
MOO	150
LOO	300
STANK	300
LTANK	450
LTUB	100
GABBY	500

4. Select products which are less than $100.00.

SELECT PROD_COD, LIST_PRICE
FROM PRODUCTS
WHERE LIST_PRICE < 100;

PROD_COD	LIST_PRICE
GNOME	10
STAND	50

5. View products which are not equal to $100.00
   SELECT PROD_COD, LIST_PRICE
   FROM PRODUCTS
   WHERE LIST_PRICE <> 100;

PROD_COD	LIST_PRICE
MOO	150
LOO	300
STANK	300

LTANK	450
GNOME	10
STAND	50
GABBY	500

Using the BETWEEN condition

Display rows based on a given range:

1. Display the product code and description from table Products which cost in between $100 and $300.

   ```
   SELECT PROD_COD,DESCRIPTION, LIST_PRICE
   FROM PRODUCTS
   WHERE LIST_PRICE BETWEEN 100 AND 300;
   ```

PROD_COD	DESCRIPTION	LIST_PRICE
MOO	Medium Cattle Trough	150
LOO	Large Cattle Trough	300
STANK	Small Septic Tank	300
LTUB	Laundry Tub	100

2. What are the products which falls in the range of $300 to $500.

   ```
   SELECT PROD_COD,DESCRIPTION, LIST_PRICE
   FROM PRODUCTS
   WHERE LIST_PRICE BETWEEN 300 AND 500;
   ```

PROD_COD	DESCRIPTION	LIST_PRICE
LOO	Large Cattle Trough	300
STANK	Small Septic Tank	300
LTANK	Large Septic Tank	450
GABBY	Football Player Statue	500